PSfMO

Friday, February 12, 2021 11:17 AM

Factorization based sfm with object priors

A set of images $f = 1 \cdots F$

A set of objects $i = 1 \cdots O$

Inscribe an ellipse D_{fi} for each bbox detection.

The aim is to find the 3D ellipsoids E_i whose projections onto the image plane best fit D_{fi}

Each ellipse is

$$u^T D_{fi} u = 0 \quad (1)$$

$u \in R^3$ is 2D point in Conic

Ellipsoid in 3D is

$$X^T E_i X = 0 \quad (2)$$

$X \in R^4$ is a 3D point.

dual quadric is defined by $Q_i = adj(E_i)$. $C_{fi} = adj(D_{fi})$

This work uses orthoghraphic camera projection.

$$P_f = \begin{bmatrix} R_f & t_f \\ 0_3^T & 1 \end{bmatrix} = \begin{bmatrix} P_{11} & \cdots & P_{14} \\ P_{21} & \cdots & P_{24} \\ 0 & \cdots & 1 \end{bmatrix}$$

$R_f \in R^{2\times 3}$, $R_f R_f^T = I_{2\times 2}$.

C_{fi}, Q_i are defined by a scale that can be fixed by setting the elements (3,3) of C_{fi} and (4,4) of Q_i to -1.

$$C_{fi} = P_f Q_i P_f^T \quad (5).$$

To recover Q_i from $\{C_{fi}\}_{f=1\cdots F}$, we need to re arrange (5) into a linear system.

Define $v_i = vech(Q_i)$ and $C_{fi} = vech(C_{fi})$.

The vech serialises the elements of the lower triangular part of a symmetric matrix,

$X \in R^{n\times n}$, $x = vech(X)$, $x \in R^g$, $g = \frac{n(n+1)}{2}$

Then rearrange P_f, P_f^T into $G_f \in R^{6\times 10}$

$$G_f = Y(P_f \otimes P_f)W.$$

$\otimes$ is the kronecker product and $Y \in R^{6\times 9}$, $W \in R^{16\times 10}$,

$vech(X) = Y\,vec(X)$ $\quad X$ is some symmetric matrix.

$vec(X) = W\,vech(X)$

vec serialises all elements of a generic matrix.

Given G_f, we have:

$$C_{fi} = G_f v_i$$

Stack all frames: $C_{fi} = vech(C_{fi})$

$$C = \begin{bmatrix} C_{11} & \cdots & C_{1O} \\ \vdots & & \\ C_{F1} & \cdots & C_{FO} \end{bmatrix} = \begin{bmatrix} G_1 \\ \vdots \\ G_F \end{bmatrix} [v_1 \cdots v_O]$$

$G^T = [G_1^T \cdots G_F^T]^T$,

$V = [v_1 \cdots v_O]$ $\quad v_i = vech(Q_i)$

$\therefore$ using the dual conics only, we can find an initial, affine solution by performing SVD over C and truncating to the first 10 components

$$C = \tilde{G}\,\tilde{V}$$

Consider an axis aligned ellipsoid in 3D as a diagonal matrix:

$$U_i = diag([a_i, b_i, c_i, -1])$$

To map to world frame:

$$T_i = \begin{bmatrix} R_i & t_i \\ 0_3^T & 1 \end{bmatrix}$$

$$Q_i = T_i U_i T_i^T$$

To project to image plane

$$C_{fi} = \begin{bmatrix} R_f & t_f \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} R_i & t_i \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} a_i & 0 & 0 & 0 \\ 0 & b_i & 0 & 0 \\ 0 & 0 & c_i & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} R_f & t_f \\ 0_3^T & 1 \end{bmatrix}^T \begin{bmatrix} R_i & t_i \\ 0_3^T & 1 \end{bmatrix}^T$$

$$= P_f Q_i P_f^T$$

Every dual Conic C_{fi} can be expressed as a conic with Coordinate center (0,0), denoted with $\check{C}_{fi}$, pre- and post- multiplied by a translation matrix T_{fi}^c, same for ellipsoids:

$$C_{fi} = T_{fi}^c \check{C}_{fi} T_{fi}^{cT} \qquad Q_i = T_i^q \check{Q}_i T_i^{qT}$$

where $\check{C}_{fi} = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{12} & c_{22} & 0 \\ 0 & 0 & -1 \end{bmatrix}$, $\check{Q}_i = \begin{bmatrix} q_{11} & q_{12} & q_{13} & 0 \\ q_{12} & q_{22} & q_{23} & 0 \\ q_{13} & q_{23} & q_{33} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$

The vectorized dual conics and dual quadrics are:

$$C_{fi} = \begin{bmatrix} c_{11} - t_1^{c^2} \\ c_{12} - t_1^c t_2^c \\ -t_1^c \\ c_{22} - t_2^{c^2} \\ -t_2^c \\ -1 \end{bmatrix} \qquad v_i = \begin{bmatrix} q_{11} - t_1^{q^2} \\ q_{12} - t_1^q t_2^q \\ q_{13} - t_1^q t_3^q \\ -t_1^q \\ q_{22} - t_2^{q^2} \\ q_{23} - t_2^q t_3^q \\ -t_2^q \\ q_{33} - t_3^{q^2} \\ -t_3^q \\ -1 \end{bmatrix}$$

Notice that translation parameters t_1^c, t_2^c and t_1^q, t_2^q, t_3^q appear linearly in highlighted entries.

$\therefore$ the ellipsoid centers can be recovered from ellipses centers by decoupling them from ellipsoid shape.

Probabilistic sfm with objects.

The relationship between Q_i, C_{fi} is defined by orthographic projection $P_f \in R^{3\times 4}$.

$$P_f = \begin{bmatrix} R_f & t_f \\ 0_3^T & 1 \end{bmatrix}$$

$$C_{fi} = P_f Q_i P_f^T \quad (2)$$

For each projection, we can define a factorization problem by vectorizing (2), by defining

$$v_i = vech(Q_i)$$
$$C_{fi} = vech(C_{fi})$$

Vech vectorises the elements of the lower triangular part of a symmetric matrix.

Rearrange P_f, P_f^T:

$$G_f = Y(P_f \otimes P_f)W \quad (3).$$

$\otimes$: kronecker product.

$Y \in R^{6\times 9}$, $W \in R^{16\times 10}$: $vech(X) = Y\,vec(X)$

$vec(X) = W\,vech(X)$

where X is a symmetric matrix. ? how to obtain X?

$\therefore$ We can rewrite (2) as

$$C_i = G v_i \quad (4).$$

C_i: $R^{6\times 1}$ conics

$v_i \in R^{10\times 1}$ quadrics

$G \in R^{6\times 10}$ cameras.

To solve for (4), we can solve the translation of the ellipsoids separately, the solution provides both the center t_i^e of each ellipsoid, v_i, and Camera matrix G.

To solve for the remaining shape and orientation, the ellipsoids are registered to the origin. Denote by w_i the R^6 vectorised quadric which has been registered from the non-translated quadric v_i. It can be shown that with w_i and its corresponding centered conics C_i^r, (4) becomes

$$C_i^r = G^r w_i \quad (5).$$

C_i^r is a R^3 vector containing all centered conics for object i observed in F frames

G^r is $R^{3\times 6}$ matrix obtained from G, it consists simply in selecting the rows and columns related to the quadric shape and disregarding the ones related to translation.

SfMO multiply by the pseudo inverse of G^r to obtain the w_i.

PSfMO use probabilistic factorisation to include object prior.

Object shape and pose decomposition.

We seek to perform an optimization constrained by an apriori knowledge on the semi-axis length of each object. first we show how these lengths appear explicitly in the factorization defined by (4).

A generic ellipsoid in dual space Q^*:

$$Q^* = Z \check{Q}^* Z^T.$$

$\check{Q}^*$ is an ellipsoid centered at origin and with axes aligned to 3D coordinates,

Z is transformation.

$$Z = \begin{bmatrix} R(\theta) & t \\ 0_3^T & 1 \end{bmatrix} \qquad (7)$$

$$\check{Q}^* = \begin{bmatrix} a^2 & 0 & 0 & 0 \\ 0 & b^2 & 0 & 0 \\ 0 & 0 & c^2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Define $e \in R^6$ as $e = [\theta_1, \theta_2, \theta_3, a, b, c]^T$, we can decompose $w_i(e)$ as

$$w_i = R_i(\theta) l_i$$

where $R_i(\theta) \in R^{6\times 3}$

$$l_i = [a^2, b^2, c^2]^T.$$

Object 3D prior.

From ShapeNet, for each object i in dataset, extract a R^3 vector

$$h_i = [a, b, c]^T$$

Containing the lengths along the main axes.

These values have been normalised so that $|h_i|^2 = 1$. and thus providing the ratio of these axis.

l_i lies in a 2D space, for each object category c, use PCA to compute a 3×2 projection matrix V_c and a mean μ_c, to project each l_i to 2D.

Eg. horizontal: $\frac{x}{y}$. Vertical: $\frac{\text{diameter}}{\text{height}}$.

Then we fit a 2D Gaussian $(0, \Sigma_c)$. this Gaussian encodes our a priori information about the ratio of different dimensions of the object.

To apply the object prior to test image:

1. Select one of six possible correspondences and the prior axes. i.e. rows of V_c, μ_c. We estimate the axes length of the 6 possible config. and keep the one with higher likelihood.
2. estimate a scale z for difference between normalised axes of the prior and the non normalised axis of ellipsoids

Probabilistic PCA with shape priors.

Assume each centered conic vector C_i^r is obtained by sampling a 2-d latent random vector s_i, and an additive Gaussian noise ϵ such that

$$C_i^r = G^r R_i z_i (V_{c_i} s_i + \mu_{c_i}) + \epsilon \quad (11).$$

ϵ: $N(0, \sigma I_3)$

s_i is the hidden variable.

z_i is the latent variable accounting for the scale difference.

In E-step, we estimate $P(s_i | C_i^r)$

In M-step, we then compute the noise variance parameter σ.

The posterior over the latent variables is:

$$P(s_i, z_i | C_i^r) \propto P(C_i^r | s_i, z_i) P(s_i) \quad (12).$$

From (11),

$$P(C_i^r | s_i, z_i) = N(x_i, \sigma I_3).$$

where $x_i = G^r R_i z_i (V s_i + \mu_{c_i})$.

$P(s_i) \sim N(0, \Sigma_{c_i})$.

Skipping the constants that do not depend on the latent variables, the logarithm of the posterior is:

$$\log(P(s_i, z_i | C_i^r)) \propto \underbrace{(x_i - C_i^r)\frac{1}{\sigma} I_3 (x_i - C_i^r)}_{①} + \underbrace{s_i^T \frac{1}{\sigma} I_3 s_i}_{②}. \quad (14).$$

Term ① is the reprojection error wrt the observed conics C_i^r.

Term ② is the prior.

We use MCMC to estimate the expectation of the latent variables $\{\hat{z}, \hat{s}_i\}$

The last element to estimate is the matrix containing the ellipsoid orientation R_i.

$$\hat{\theta} = \arg\min_{\theta} \| \hat{x}_i - C_i^r \|^2 \quad (15).$$

$\hat{x}_i = G^r R_i \hat{z}_i (V \hat{s}_i + \mu)$

From the solution $\hat{\theta}$, we can compute R_i and obtain quadric estimate as:

$$\hat{w}_i = R_i(\hat{\theta}) \hat{l}_i \hat{z} \quad (16).$$

In the M-step, we use $\hat{w}_i$ to estimate noise covariance σ:

$$\hat{\sigma} = \frac{1}{3}\sum_{i=1}^{O} \{ \|C_i^r\|^2 - 2\hat{w}_i^T G^r C_i^r + trace(\hat{w}_i^T \hat{w}_i G^T G^r) \}.$$

Probabilistic PCA:

Consider a dataset $X = \{x_n\}$ of N data points, $x_n \in R^D$. We aim to represent each x_n under a latent variable $z_n \in R^K$ with lower dimension $K < D$. The set of principle axes W relates the latent variables to the data.

Assume each latent variable is normally distributed.

$$z_n \sim N(0, I).$$

The data point is generated via a projection:

$$x_n | z_n \sim N(W z_n, \sigma^2 I).$$

where $W \in R^{D\times K}$ are known as the principle axes.

We want to estimate W, σ^2.

Marginalizing out the latent variable, the distribution of each data point is

$$x_n \sim N(0, WW^T + \sigma^2 I).$$

MAP

$$P(W, Z | X) \propto P(W, Z, X).$$

MAP can find mode but does not provide insight. We next use variational inference, where the posterior distribution $P(W, Z|X)$ is approximated using a variational distribution $q(W, Z)$ parameterized by λ. The aim is to find the λ that minimizes the KL divergence between q and the posterior, $KL(q(W,Z) \| P(W, Z|X))$, or equivalentally, that max. the evidence lower bound

$$\mathbb{E}_{q(W,Z;\lambda)}[\log P(W, Z, X) - \log q(W, Z; \lambda)].$$

PPCA:

PCA is $\quad y = Q^T x$,

the jth projection is $\quad y_j = q_j^T x$.

If the projections are ranked according to variance, then the M variables that reconstruct the original data are given by $y_1, \cdots y_M$, The reconstructed data is:

$$\hat{x} = Q_{1:M}\, y_{1:M}$$

In PPCA, the transform is converted into a model by explaining the discarded variance as observation noise:

$$x = Wy + e. \qquad e \sim N(0, \sigma^2 I)$$

Define S as the sample covariance matrix, and

$$C = WW^T + \sigma^2 I.$$

Then the log likelihood of data is

$$\log p(x) = -\frac{Nd}{2}\log 2\pi - \frac{N}{2}\log|C| - \frac{N}{2}Tr(C^{-1}S).$$

where N is number of data.

B1. PPCA

PCA:

The goal of PCA is to find a linear mapping from $y \in R^R$ to a lower dimensional latent code $z \in R^Q$, that captures as much variance as posible.

Considering a 1D latent space $z \in R$ $(Q=1)$, we are looking for a vector $u \in R^R$ that max. the variance captured in z:

$$Var[z] = Var[u^T y].$$

Define $\quad \mu = \frac{1}{M}\sum_{m=1}^{M} y_m$

$$\therefore \quad \mathbb{E}[u^T(y-\mu)(y-\mu)^T u] = u^T \Sigma u.$$

We require $\|u\|_2 = 1$ and can choose it as the eigenvector corresponding to the largest eigenvalue λ,

$$u^T \Sigma u = \lambda u^T u = \lambda.$$

and $Var[z]$ is max.

The above idea can be generalized to Q-dimensional latent spaces, then the linear mapping $U \in R^{R\times Q}$ is found by computing the eigenvalue decomposition of Σ:

$$\Sigma = V \Lambda V^T.$$

let $U = V_Q := [v_1 \cdots v_Q]$, yields the linear mapping max. the variance and thereby also minimizing the reconstruction error.

The outline from the above example can be stated more precise in terms of an encoding transformation and a decoding transformation, which will correspond to the recognition model and generative model in PPCA:

Given data $Y = \{y_1, \cdots, y_M\} \subseteq R^R$, with mean μ and covariance Σ, define

$$U = V_Q := [v_1 \cdots v_Q] \in R^{R\times Q}$$

using the first Q eigenvectors $v_1 \ldots v_Q$. obtained from $\Sigma = V \Lambda V^T$. Then, PCA defines an encoding transform

$$z = U^T(y - \mu)$$

and a decoding transform

$$\tilde{y} = Uz + \mu.$$

The problem with PCA is no generative model. PPCA wraps a probabilistic interpretation around a linear encoding and decoding. We still assume a linear model:

$$y = Uz + \mu + \epsilon.$$

$\epsilon \sim N(\epsilon; 0, \sigma^2 I_R)$

$U \in R^{R\times Q}$.

$P(z) = N(z; 0, I_Q)$.

this implicitly defines a generative model:

sample $z \sim p(z)$.

$$y \sim p(y|z) = N(y; \mu, Uz + \mu, \sigma^2 I_R)$$

We also need the recognition model, i.e. the posterior $P(z|y)$.; representing the encoding transformation:

$$P(z|y) = N(z; S^{-1}U^T(y-\mu), \sigma^{-2}S^{-1}).$$

where $S = U^T U + \sigma^2 I_Q$.